COMPLEX INTELLIGENT MACHINES

Herschel B. Smartt, Ph.D., Charles R. Tolle, Ph.D.,

Kevin L. Kenney

Idaho National Engineering and Environmental Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-2210

ABSTRACT

The machine control problem is normally approached from the perspective of having a central body of intelligence (and control) in the machine [Albus, 1991]. However, we present a conceptual design of a machine using distributed learning and intelligence. This new design is loosely based on biological models of social insects. For example, in an ant colony each ant functions according to local rules of behavior [Hölldobler and Wilson, 1990, see chapters 8 and 9]. There is no “king” or “queen”, although the latter name has been given to the reproducing ant. Following a similar approach, we present a modular machine architecture in which each machine element has local rules of behavior (and local learning) along with a global element that influences local behavior (but does not dictate actions). A prime goal is to develop methods of learning and behavior modification that ensure global stability and optimization of the total machine; we discuss the theoretical aspects of ensuring such optimal performance.

INTRODUCTION

James Albus [1991] at NIST has defined machine intelligence as “the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that supports the system’s ultimate goal.” Following Albus’ intent, we can say that intelligent machines are those that either know or can learn everything they need to know to perform a process or task. Such machines may be able to perform a process or task autonomously (without operator intervention) or semi-autonomously (with operator intervention).

In this paper, we present a conceptual design of a machine using distributed learning and intelligence. Related work has been conducted, for example, by Dorigo and Colorni [1996] using ant-

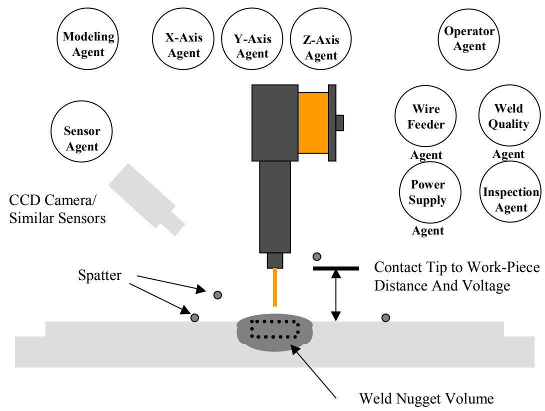

Figure 1. Arc spot welding machine with agents for the power supply, electrode wire feeder, positioner axes, sensor, and operator interface.

based local behavior of multiple agents to solve the Travelling Salesman Problem and other classical hard problems. Schatz et al. [1999] formulated a model for route learning in ants. Lambrinos et al. [2000] used a similar model for navigation of a mobile robot. Overgaard, Petersen, and Perram [1995, 1996] used local agent control of dynamic motion and path planning in multiple link robot arms.

Consider an intelligent machine in which various machine functions are carried out in a distributed manner. A schematic of such a machine for arc spot welding is shown in Figure 1. In addition to the machine hardware required (most of which is not shown) there are several "agents". These agents have local control of various machine functions and are able to communicate with each other and with an operator agent, see Figure 2. The operator agent may be a human or may be an interface to a human (or even an interface to another machine). (Although it would be possible to focus on autonomous machines, we chose not to do so; our machines will interact with humans who have supervisory control authority.)

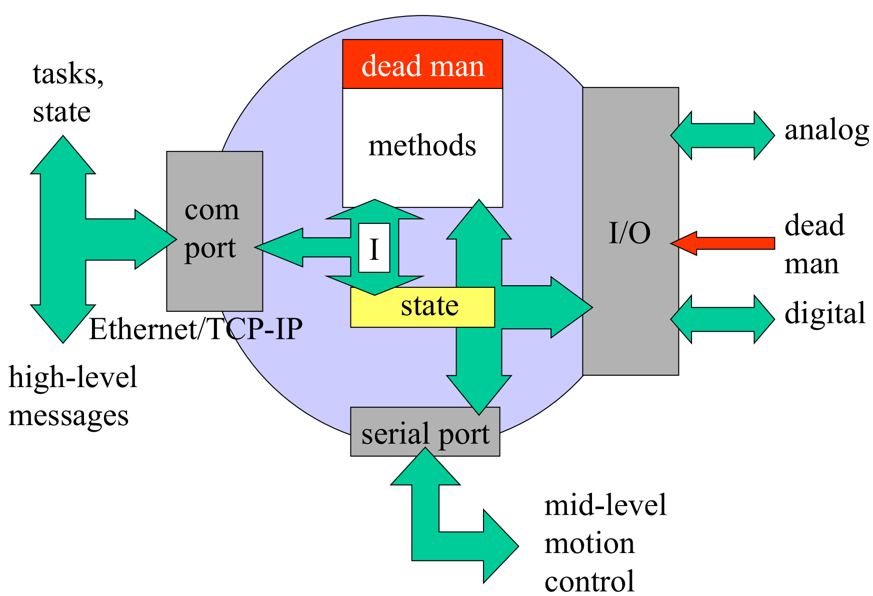

Figure 2. Agent block diagram.

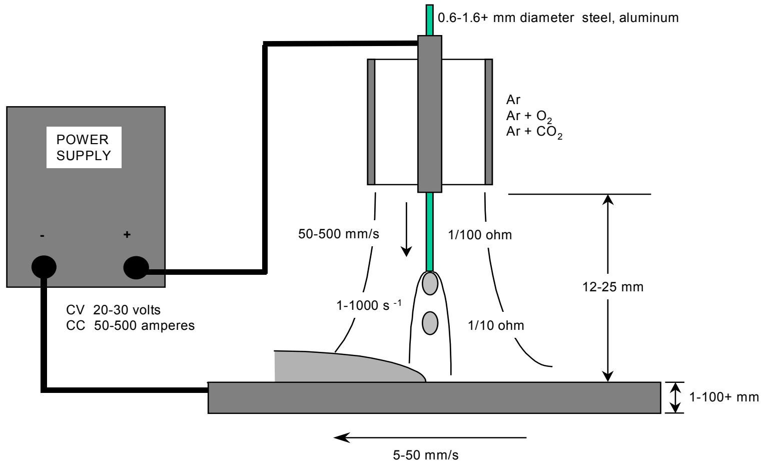

Figure 3. Schematic of gas metal arc welding process showing typical values of parameters.

The various agents will incorporate knowledge of how to perform tasks, the ability to learn from experience, and memory of past performance. The agents will also be able to optimize both their local behavior and the global behavior of the total machine.

To formulate such a machine, we need a variety of methods. In addition to distributed learning and control, we also chose to have our machines learn rules of behavior. This is distinct from learning control trajectories, a method frequently employed for machine learning. Our rules will be embodied using a variant of fuzzy logic [Johnson and Smartt, 1995] that allows the system to learn by back propagation [Rumelhart, 1986]. However, we discuss the application of iterative learning control[a] to distributed intelligence. Iterative learning control is a recent set of methods for learning control trajectories that is well suited to iterative processes. However, iterative learning control methods may also be used to learn the weighting of rules for local optimization. We also discuss a new method of global optimization that uses artificial neural networks that learn the contribution of local behaviors to global cost.

SIMPLE INTELLIGENT WELDING MACHINE

Now consider the welding machine control problem. This is a much more complicated problem than two-dimensional motion control. First, there is a motion control problem involved. Even simple automated welding machines may have three degrees of motion. Consider welding in the flat position (e.g. joining two flat plates edge to edge with the plates in the horizontal plane). The welding torch must move along the weld joint. It also needs to be able to move at right angles to the weld joint (in the horizontal plane) to track misalignment of the joint with the axis of primary motion. Finally, it needs to move in a vertical direction to obtain changes in the contact tube to weldment distance. In addition, the weld torch may be mounted with a lead or lag angle relative to the weld joint. That is, the torch may be nominally vertical to the plates, but tilted backwards or forwards, respectively, with respect to the welding direction. For other weldment configurations, the torch may be leaned to one side or the other. Finally, the torch may be moved laterally with respect to the weld joint in a weaving pattern to effectively increase the width of the weld bead. Robotic welding systems may be even more complicated.

---

a. Uchiyama, 1978; Arimoto, Kawamura, and Miyazaki, 1984; Moore, 1993

Welding also involves selection of proper values of the process independent variables, see Figure 3. Disturbances to the process or uncertainties in the welding conditions may result in a need for the welding process independent variables to be changed during welding. Consequently, we need to consider that the trajectory we must obtain involves multiple degrees of motion via the robot as well as multiple degrees of motion through welding parameter space. What we seek is a set of generic rules that will ensure that the weld is made in some manner that will result in a structurally reliable weldment. Further, we want the welding machine to tune those rules to obtain a more robust process than would result from a fixed set of rules.

Consider a specific welding control problem. We desire to fabricate a steel structure using arc spot welding. Thus, steel sheet will be welded to an underlying structure by means of weld nuggets deposited into circular holes in the sheet. This geometry is shown in Figure 4.

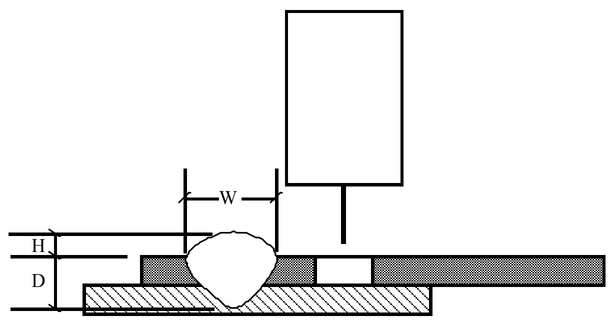

Figure 4. Cross section of gas metal arc spot weld showing a hole in the top sheet and a completed weld bead.

In this situation, the weld torch may be moved to a suitable position over a weld site, using motion control as discussed earlier. The welding power supply contactor is activated, the power supply voltage is set, the shielding gas is turned on, and the electrode wire is fed downward. This will result in ignition of an arc with corresponding heat and mass transfer to the weldment. After a suitable time, the power supply contactor is deactivated and the electrode wire feed is stopped. A short time later the shielding gas is turned off. Although this is perhaps the simplest arc welding example we can consider, there are still important control decisions that ensure that the weld will meet its acceptance requirements.

A good weld in this example is one that is strong enough, does not excessively over or under fill the hole, has minimal spatter, and does not contain gross defects such as cracks or porosity that could lead to failure. To be strong enough, the weld bead must adequately penetrate the lower structure (but not excessively melt through that structure) while fusing into the upper sheet. For most applications, the cross-sectional area of the weld bead in the plane of the interface between the upper sheet and the lower structure needs to be equal to or greater than some critical amount.

To obtain a good weld in this example, the current must be high enough but not too high and the weld time (the time the arc is on) must be equal to or greater than some critical minimum. This will ensure that adequate heat and mass have been transferred to the weldment. It is also necessary for the voltage to be above some minimum (to reduce spatter) and below some maximum (to avoid melt through and burn back).

GLOBAL OPTIMIZATION

One of the key challenges to achieving intelligent distributed learning with global optimization within our welding machine is the interaction between the global optimization cost function and local agent optimization cost functions. Each independent optimization agent must be able to make changes to its locally controlled parameters while keeping in mind its effects on the global cost/process. Traditionally, industrial process optimization is broken down into sub-components then optimized locally. If there is time and resources, once the sub-components have been optimized, an engineering team is formed to globally optimizing the interactions between sub-components through manual trial and error adjustments away from the sub-components optima. Today there is a growing interest in global process self-optimization, or near optimization, through the use of "swarm intelligence" [Bonabeau et al., 2000]. One method in particular is based on how ants function in nature while searching for food [Dorigo et al., 1996]. Each ant acts as an independent agent making random decisions on where to search with each move. As traffic increases along a particular path, ants crossing that path will be biased with a greater probability to follow the already more traveled path by its higher pheromone level. This tendency is reinforced as ants travel back and forth along the path between food and the anthill faster than other ants on competing paths. This leads to a faster increase in the pheromone trail on the shorter paths with respect to the longer competing paths, which attracts additional ants to the short paths. This is a never-ending reinforcement of the global optimization cost function, i.e. increased food movement back to the anthill. However, as is pointed out by Bonabeau et al. [2000], there is also a kind of integral wind up effect in ant behavior. When a shorter path is introduced after the "best path" has been found, the system has a hard time finding it unless the dynamics are changed, e.g. the initial food source is used up. This limitation can be overcome by modeling the pheromone trial as evaporative [Bonabeau et al., 2000; Dorigo et al. 1996]. It is important to make clear the key ideas being presented within this optimization system: each ant adds its own piece of cost to the global cost function (food delivered), and they are able to communicate to each other about their success (pheromone trail).

A slightly different intelligent distributed learning system can be expressed in human terms as an everyday project team. Here each team member is an individual agent that contains a wide range of experience, talents, and education. Normally, such teams have a team/project leader whose role is key to their achieving intelligent distributed learning on a global optimization problem. The team leader, and his allowed interactions, differentiates this method of distributed learning from the ant's. Within this model of distributed learning, the global cost function is contained within the team leader and acts as an agent of its own. Tasks are distributed among team members as well as sub-groups of team members. These agents progress in solving their tasks, as well as developing localized communication paths between agents, i.e. real time reconfiguration of the sub-groups. More importantly, the key concept within this structure is that the team leader cannot dictate any particular action to a team member. In human terms, this is primarily due to the team leader's lack of technical details and/or conceptual understanding required to solve any particular subtask. Remember, the team leader is globally oriented. However, team leaders can attempt to influence a particular team member's actions in order to achieve the global optimal solution. For example, the team leader can inform a member/agent that by increasing the tolerances within their portion of the process, all of the remaining process can speed up dramatically, i.e. the other agents are waiting on that particular agent due to the extra time required to optimize his subtask, even though it will not add much to the global cost function. We propose that this interaction between the team leader and individual team members is the key to the successful development of an intelligent global distributed learning algorithm, as opposed to "swarm intelligence." Furthermore, we differentiate an intelligent global distributed learning algorithm from a centralized learning algorithm, such as traditional neural networks, by not allowing any agent to dictate to another agent its actions, i.e. the team leader is not allowed to force any agent into a particular action. In short, an intelligent global distributed learning algorithm must allow each agent its own localized cost function and the ability to solve its subtasks primarily by itself with non-dictated feedback from the other agents. Global optimization is actually achieved through the local optimization procedures employed by each agent while taking into account the global effects of its choices.

Using the human project team model just discussed, consider the intelligent spot welding machine outlined in Figure 1. This machine has a team leader, the Weld Quality Agent. Its job is to evaluate the overall success of welds produced by the machine and supply feedback to the local agents—i.e., x-axis agent, y-axis agent, z-axis agent, power supply agent, wire feeder agent—on their effectiveness in producing quality welds. The key to designing a particular algorithm is how the team leader, the Weld Quality Agent, is allowed to interact with each of the local agents. A further complication is by what methods will the team leader pass localized global cost information to each agent.

With this in mind, we consider the following initial architecture and algorithm for study, as outlined in Figure 1. This algorithm is based on a global cost function maintained and calculated within the Weld Quality Agent using generic weld parameter information developed from machine's agents.

$$Weld\_Quality = V_Q(V_D, V_M) + S_Q(S_I, V_I) + B_Q(S_I, V_I) + M_Q(P_I) + T_Q$$

where

V_Q represents the quality of the well nugget volume based on the desired volume, V_D, and the measured/calculated volume, V_M.

S_Q represents the quality cost of spatter produced during the welding process based on sensor inspection, S_I, and operator visual inspection, V_I.

B_Q represents the quality cost associated with burn back.

M_Q represents the quality of the mechanical joint produced by the machine based on the operator's physical inspection, P_I.

T_Q represents the cost to quality due to the time involved in producing the weld.

Now that the form of the global cost function has been chosen, the next step is to define the relationship and method for communicating global cost information to the local agents. We propose to accomplish this task by adding these effects of the localized global cost, C_{agent}^{global}, to the traditional local cost function, C_{agent}^{local}.

$$C_{agent} = \alpha C_{agent}^{local} + (1-\alpha) C_{agent}^{global}$$

The α term is used to adjust the balance between local cost and global cost variations on the local optimization process. It is planned that for new welding setups, the machine may fix α to be 1 for an initial period of time, thereby allowing development of the initial relationships between the local agent costs and global costs before proceeding with the augmented local cost function above (i.e. $\alpha \neq 1$). Note that the effect due to the global cost is based on multiple terms within the weld quality cost function, e.g. $C_{agent}^{global}(V_Q, S_{QI})$. These mappings form the uniqueness and key difficulty of the proposed algorithm;

namely, how will the algorithm obtain a mapping from global cost effects to local cost effects, i.e. $Weld\_Quality \rightarrow C_{agent}^{global}$? To simplify the initial algorithm and its solution, it will be assumed that the effect of the global cost on a particular agent is a linear combination of each of the $Weld\_Quality$ sub-terms:

$$C_{agent}^{global}(V_Q, S_Q, B_Q, M_Q, T_Q) = a_1 V_Q + a_2 S_Q + a_3 B_Q + a_4 M_Q + a_5 T_Q$$

where the a_i's are constants.

With this assumption in hand, it is planned to learn the forward direction map, from the local agent costs to the global cost function terms, via a neural network mapping:

$$(C_{WF}^{local}, C_{XA}^{local}, C_{YA}^{local}, C_{ZA}^{local}, C_{PS}^{local}) \xrightarrow{neural\_network} M_Q$$

where C_{WF}^{local} represents the wire feeder agent's cost, C_{iA}^{local} represents the i^{th}-axis agent's cost, C_{PS}^{local} represents the power supply agent's cost, and as above the a_i's are constants. This forward neural network mapping will then be used to produce the reverse gradient mapping by way of the back-propagation training method. In addition to training the neural network based on the traditional error prediction feedback of the forward network, the change in each of the global cost terms (ΔM_Q) due to changes in the local agents $(\Delta C_{WF}^{local}, \Delta C_{XA}^{local}, \Delta C_{YA}^{local}, \Delta C_{ZA}^{local}, \Delta C_{PS}^{local})$ will be back propagated through the network all the way to its inputs (this process is not used to update weights). By doing this, one is attempting to use the back-propagation training method to relate a change in global cost to the local agents by exploiting the gradient knowledge contained within the forward mapping of the neural network. We are not attempting to reverse map the input to outputs, instead we are only trying to obtain gradient information at the input of the neural network based on the change of the output of the network. In fact, the back-propagation algorithm is based on a gradient descent method, which back-propagates gradient information about the error in the outputs due to the inputs in a similar fashion.

It is planned to use radial basis neural networks within this part of the project because of their connection to Takagi-Sugeno fuzzy systems. This connection will be used to obtain a qualitative understanding of the mapping between the local agent costs and the global sub-cost. This is possible because radial basis neural networks and Takagi-Sugeno fuzzy systems have been shown to be mathematically the same, though developed from a different understanding [Spooner and Passino, 1996]. In essence, it is thought that one can develop a fuzzy model of the mapping process from local costs to global sub-costs by using a radial basis neural network [Passino, 1999]. This qualitative understanding of the relationships between local costs and global sub-costs can then be used in future model development for the welding process as well as in more traditional control systems for welding processes.

CONCLUSION

An approach to design of an intelligent machine has been presented based on distributed intelligence. Local agents are used to control individual machine functions and to process information needed by the machine functions. Examples of how this approach may be used to build a specific machine are presented for an arc spot welding application. A possible agent internal structure is presented that provides for local rules of behavior and safety considerations. An initial method for accomplishing distributed learning with global optimization has been presented. The learning method outline within this paper will form the basis for our continued research.

ACKNOWLEDGEMENT

This work is supported by the Basic Energy Sciences, Office of Science, U.S. Department of Energy, under DOE Idaho Operations Office Contract DE-AC07-99ID13727.

REFERENCES

Albus, James. S., “Outline for a Theory of Intelligence,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol 21, no. 3, May/June 1991, pp. 473-509.

Arimoto, S., S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning," Journal of Robotic Systems, Vol. 1, pp. 123-140, March 1984.

Bonabeau, Eric; Théraulaz, Guy. “Swarm Smarts,” *Scientific American*, Mar. 2000, pp.72-79.

Dorigo, Marco; Maniezzo, Vittorio; Colorni, Alberto. “Ant System: Optimization by a Colony of Cooperating Agents,” IEEE Transctions on Systems, Man, and Cybernetics-PartB: Cybernetics, v 26, n 1, Fed. 1, 1996, pp. 29-41.

Hölldobler, B. and Wilson, E. O., *the ANTS*, The Belknap Press of Harvard University Press, Cambridge, Mass.,1990.

Johnson, J. A., and H. B. Smartt, "Advantages of an Alternative Form of Fuzzy Logic," IEEE Transactions on Fuzzy Logic, 3, pp. 149-157, 1995.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., and Wehner, R., “A mobile robot employing insect strategies for navigation”, Robotics and Autonomous Systems, 30 (2000) 39-64.

Moore, K. L., Iterative Learning Control for Deterministic Systems, Series on Advances in Industrial Control, Springer-Verlag, 1993.

Overgaard, L., Petersen, H. G., and Perram, J. W., “A General Algorithm for Dynamic Control of Multilink Robots”, The International Journal of Robotics Research, Vol. 14, No. 3, June 1995, pp. 281-294.

Overgaard, L., Petersen, H. G., and Perram, J. W., “Reactive Motion Planning: a Multiagent Approach”, Applied Artificial Intelligence, 10:35-51, 1996.

Passino, Kevin. “Intelligent Control,” Workshop S-5, 38th IEEE Conference on Control, Phoenix, AZ, Dec. 7-10, 1999.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, ed. D. E. Rumelhart and J. L. McClelland, MIT Press, Cambridge, MA, pp. 318-362 (1986).

Schatz, B., Chameron, S., Beugnon, G., and Collett, T. S., “The use of path integration to guide route learning in ants”, Nature, Vol. 399, No., 24, June 1999, pp. 769-772.

Spooner, Jeffrey; Passino, Kevin. “Stable Adaptive Control Using Fuzzy Systems and Neural Networks,” IEEE Transactions on Fuzzy Systems, v 4, n 3, Aug. 1996, pp. 339-359.

Uchiyama, M., "Formation of High Speed Motion Pattern of Mechanical Arm by Trial," Transactions of the Society of Instrumentation and Control Engineering, Vol. 19, pp. 706-12, May 1978.